

CLAIMS

1. A method for replicating a software application in a multi-computer architecture (cluster), whereas said software application may be executed
5 beforehand on a first computer of said cluster forming a primary node and intended for replication on at least one other computer of said cluster forming a secondary node, comprising a replication of the resources associated with said software application, characterised in that it includes on the flow updating of the replicated resources by a dynamic introspection
10 mechanism supplying the structure of the application to be replicated, as well as a dynamic graph of the resources and dependencies implemented.
2. A replication method according to claim 1, characterised in that it includes a creation and a maintenance of a dependency tree, supplying at
15 all times information on the resources which ought to be replicated.
3. A replication method according to one of the claims 1 or 2, characterised in that it includes a checkpointing mechanism via which the resources to be replicated are replicated on one or several secondary nodes.
20
4. A replication method according to claim 3, characterised in that it includes three steps:
 - capturing resources on the primary node,
 - transfer over the network towards one or several secondary nodes,
 - 25 - restoration on the secondary node(s).
5. A replication method according to any of the previous claims, characterised in that the replicated resources include:
 - the virtual memory of each process affected as well as its calling
30 stack,
 - system resources (inter-process communication, network connection, etc.) and
 - data written on disks.

6. A replication method according to any of the previous claims and claim 3, characterised in that it includes a mechanism for optimising the checkpointing mechanism.
- 5 7. A replication method according to claim 6, characterised in that the checkpointing mechanism is incremental.
8. A replication method according to one of the claims 6 or 7, characterised in that the checkpointing mechanism is discriminating.
- 10 9. A replication method according to one of the claims 6 to 8, characterised in that the checkpointing mechanism includes at least one of the following functions:
- a process synchronisation barrier (PSB),
 - 15 - a resource management (RM),
 - a system resources management (SRM), and
 - a process resources management (PRM).
- 20 10. A replication method according to any of the previous claims, characterised in that it includes moreover a mechanism for replicating applicative data files between an operational node (OP) whereon the application is run and a so-called stand-by node (SB).
- 25 11. A method ensuring functional continuity of a software application in a multi-computer architecture (cluster), said application being executed at a given time on one of the computers of the cluster, called primary or operational node, while the other computers of said cluster are called secondary, said process implementing the replicating process according to any of the previous claims,
- 30 characterised in that it includes the following steps:
- replication of the application on at least one of the secondary nodes, in order to provide at least one clone of said application,
 - on the flow updating of said clone(s), and

- when detecting a fault or an event affecting said operational node, switching the service towards one at least of said clones.

12. A functional continuity method according to claim 11, characterised in
5 that the replication of the application is of holistic nature.

13. A functional continuity method according to any of the claims 11 or 12, characterised in that it includes moreover updating the clones of the application.

10

14. A functional continuity method according to any of the claims 11 to 13, characterised in that it includes moreover supervising the state of the resources necessary to the operation of the application.

15 15. A functional continuity method according to any of the claims 11 to 14, characterised in that it further includes, when detecting a fault or an event affecting said operational node, a step for electing, among the clones installed on secondary nodes, a clone to be substituted for the initial application, whereas the node whereon said clone elect is installed becomes
20 the new operational node.

16. A functional continuity method according to any of the claims 11 to 15, characterised in that it includes moreover a record on each clone of messages received by the primary or operational node, said messages
25 being re-injected into the clone elected as new primary when switching.

17. A multi-computer system designed for running on at least one of said computers at least one software application, implementing the method ensuring functional continuity according to any of the claims 11 to 16.

30

18. An application of the replicating method according to any of the claims 1 to 10, for automatic optimisation of the information-processing resources by load sharing by dynamic process distribution.

19. An application of the replicating method according to any of the claims 1 to 10, for non-interruptive maintenance by process re-location upon request, over a data-processing resource network.
- 5 20. An application of the replicating method according to any of the claims 1 to 10, for preservation of applicative context in mobile applications.